DETAILED ACTION

RESPONSE TO ARGUMENTS

1. Applicant's arguments filed 03/30/2010 have been fully considered but they are not persuasive. Currently, claims 2, 11, 29-33 and 42 are canceled, claims 17-28, 34-40 and 43-44 are withdrawn, and claims 1, 3-10, 12-16 and 41 are pending for examination.

2. In response to applicant's arguments with regard to the independent claim 1 rejected under 35 U.S.C. 103(a) that Rao teach away from the claimed invention because Rao only specifies the use of a named executable (i.e. utilizing Sync ML DM protocol col. 8, ll. 65-67); applicant's arguments have fully been considered, but are not found to be persuasive.

The examiner respectfully disagrees, because Rao's column 8, lines 65-67 does not disclose "specifies the use of a named executable," as Rao only discloses the use of SyncML DM protocol; and if applicant's were to equate the use of SyncML DM protocol to "specifies the use of a named executable," then wouldn't applicant's own application, which utilizes the SyncML DM protocol/commands also be equated to specifying "the use of a named executable"? In summary, Rao does not teach away from application, as Rao teaches the utilization of an enhanced SyncML DM command, and not the standard SyncML DM command.

Please note that one cannot show nonobviousness by attacking references individually where the rejections are based on combinations of references. See *In re Keller*, 642 F.2d 413, 208 USPQ 871 (CCPA 1981); *In re Merck & Co.*, 800 F.2d 1091, 231 USPQ 375 (Fed. Cir. 1986).

3. In response to applicant's arguments with regard to the independent claim 1 rejected under 35 U.S.C. 103(a) that the combination of the references is improper because DTD does not describe any implementation of Sync ML technology or discuss any executable, and Szeto does not help because Szeto relates to technique for messages received from different environment; applicant's arguments have fully been considered, but are not found to be persuasive.

The examiner respectfully disagrees, because DTD is directly associated with Sync ML technology, as the top the DTD documentation discloses "SyncML Meta Information DTD," and Rao discloses the use of an executable (e.g. module operating/executing on the firmware update data) and Szeto teaches the unidentified executable (e.g. unidentified supporting application).

Please note that one cannot show nonobviousness by attacking references individually where the rejections are based on combinations of references. See *In re Keller*, 642 F.2d 413, 208 USPQ 871 (CCPA 1981); *In re Merck & Co.*, 800 F.2d 1091, 231 USPQ 375 (Fed. Cir. 1986).

4. In response to applicant's arguments with regard to the independent claim 1 rejected under 35 U.S.C. 103(a) that the combination of the references does not teach/suggest every claimed feature because Szeto teaches IM application as the actual data (i.e. movie trailer and not executable) and supporting application is the executable (i.e. media player) and applicant disagrees the examiner's assertion that IM application is software for implementing an instruction set, and the supporting application is not a required application, since supporting application is needed only when the IM application is unable to render the corresponding data; applicant's arguments have fully been considered, but are not found to be persuasive.

The examiner respectfully disagrees, to further clarify the examiner's assertion, IM application is an application/software/executable that is executed to enable instant messaging between users (i.e. IM application is the application/software/executable that enables the instant messaging environment), and when the transferred instant message include data (e.g. media player data) that IM application is unable to render, such as data that require a supporting application (e.g. media player), the supporting application is then called upon for rendering the corresponding data (e.g. media player data). In summary, IM applicant and supporting application are both executable, wherein the IM application is the executable for processing instant messages and the supporting application (e.g. media player) is the executable for corresponding data (e.g. media player data) that the IM application is unable to process; to further explain, the received IM application data initiates the operation of an unidentified executable, because only upon further examination of the received IM application data can the unidentified executable (e.g. unidentified supporting application such as media player) be determined and called upon for rendering the corresponding data (e.g. media player data).

5. In response to applicant's arguments with regard to the independent claim 1 rejected under 35 U.S.C. 103(a) that the combination of the references does not teach/suggest every claimed feature because Szeto teaches the execution is commanded by the receiving client, as Szeto teaches (col. 12, ll. 49-53) the IM environments evaluate messages received by handlers in the IM clients and it is the IM environments that determines an appropriate action for user and IM messages; therefore, Szeto teaches the execution is identified by the sending client; applicant's arguments have fully been considered, but are not found to be persuasive.

The examiner respectfully disagrees, and in accordance to applicant's reasoning, the message received by the client is send from the sender, wherein the sender constructs the corresponding message including the information/instruction/command; therefore, it would be the sender that informs/instructs/commands the execution at the receiving client, because it is the sender that constructs the message having the corresponding information/instruction/command to be transferred to and received by the client. Additionally, Szeto does teach the execution is identified by the receiving client, because based on the identifier the receiving client identifies which supporting application is needed for execution.

6. In response to applicant's arguments with regard to the independent claim 1 rejected under 35 U.S.C. 103(a) that the examiner's analysis is flawed because Szeto has no instruction and since there is no instruction, the execution in Szeto cannot reasonable be said to be commanded by the sending client; applicant's arguments have fully been considered, but are not found to be persuasive.

The examiner respectfully disagrees, because the examiner is not certain where in Szeto teaches that there is "no instruction", and therefore is unable to properly respond to applicant's arguments.

7. In response to applicant's arguments with regard to the independent claim 1 rejected under 35 U.S.C. 103(a) that the combination of the references does not teach/suggest the claimed feature "automatically determining, from metadata of the first data, a content type of the first data at the receiving client" because Szeto teaches the content type is determined at the sending client; additionally, Szeto does not command an executable; applicant's arguments have fully been considered, but are not found to be persuasive.

Please note that the features upon which applicant relies (i.e., automatically determining at the receiving client) are not recited in the rejected claim(s). Although the claims are interpreted in light of the specification, limitations from the specification are not read into the claims. See *In re Van Geuns*, 988 F.2d 1181, 26 USPQ2d 1057 (Fed. Cir. 1993). Additionally, Szeto does teach the content type is determined at the receiving client, as the receiving client makes the determination via the identifier.

Furthermore, Rao discloses the use of an executable (e.g. module operating/executing on the firmware update data) and Szeto teaches the unidentified executable (e.g. unidentified supporting application).

8. In response to applicant's arguments with regard to the independent claim 1 rejected under 35 U.S.C. 103(a) that the combination of the references does not teach/suggest every claimed feature because Szeto teaches the receiving client is told what excitable is to be used; applicant's arguments have fully been considered, but are not found to be persuasive.

The examiner respectfully disagrees, because Szeto determines what executable to use based on the received identifier, similar to applicant's receiving client determining what executable to use based on the received metadata.

9. In response to applicant's arguments with regard to the independent claim 1 rejected under 35 U.S.C. 103(a) that Sezto is not pertinent to the particular problem with which applicants are concerned because Sezto does not identify the corresponding application (e.g. supporting application) using metadata as in DTD, instead Sezto identifies the supporting application using an identifier specified by a sending IM client; and Sezto's sending client is not telling the receiving client whether and what to do with the identified data; applicant's arguments have fully been considered, but are not found to be persuasive.

Please note one cannot show nonobviousness by attacking references individually where the rejections are based on combinations of references. See *In re Keller*, 642 F.2d 413, 208 USPQ 871 (CCPA 1981); *In re Merck & Co.,* 800 F.2d 1091, 231 USPQ 375 (Fed. Cir. 1986). Furthermore, the examiner relied on Rao and "SyncML Meta-Information DTD" for the teaching of using metadata as in DTD and relied on Sezto for the teaching of identify the corresponding application (e.g. supporting application) using the identifier, wherein the identifier is functionally equivalent to the metadata; therefore, the resulting combination of the references further teaches using the metadata/identifier to identify the corresponding application (e.g. supporting application). Additionally, Sezto's sending client is telling the receiving client whether and what to do with the identified data via the identifier provided by the sending client.

I. ACKNOWLEDGEMENT OF REFERENCES CITED BY APPLICANT

10. The Information Disclosure Statement(s) has been reviewed by the examiner and is found to comply with the provisions of 37 CFR 1.97, 1.98 and MPEP § 609.

II. ELECTION/RESTRICTION

11. This application contains claims 17-28, 34-40 and 43-44 drawn to an invention nonelected with traverse in the reply filed on 08/17/2009. A complete reply to the final rejection must include cancellation of nonelected claims or other appropriate action (37 CFR 1.144) See MPEP § 821.01.

III. REJECTIONS BASED ON PRIOR ART

Claim Rejections - 35 USC § 103

The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

12. Claims 1, 3-10, 12-16 and 41 are rejected under 35 U.S.C. 103(a) as being unpatentable over Rao et al. (US Patent 6,978,453) in view of "SyncML Meta-Information DTD" and Szeto (US Patent 7,188,143).

13. As per claim 1, Rao teaches a method comprising:

causing, at least in part, receiving at an electronic device (Fig. 1, ref. 107) a command (e.g. update command) specifying execution on first data (Fig. 1; col. 3, ll. 21-44; col. 6, l. 49 to col. 7, l. 19 and col. 8, l. 25 to col. 12, l. 19), wherein the first data executed to be associated with firmware update data;

utilization of metadata protocol, wherein the first leaf node would have the corresponding metadata (col. 6, l. 49 to col. 7, l. 19);

automatically determining (e.g. determining via recognition) a property of the identified first data (e.g. property identifying first data to be firmware update data) (col. 8, l. 25 to col. 12, l. 19), as the received command is recognized by the electronic device to have the property associated with firmware updating;

causing, at least in part, operating on the identified first data using an executable (e.g. module) (col. 5, ll. 23-32 and col. 5, l. 61 to col. 6, l. 4), as the module would operate on the firmware update data via downloading and updating processes.

Rao does not teach the method comprising: an unidentified executable; determine content type from the metadata; and automatically identifying an executable using the content type determined from the metadata.

SyncML Meta-Information DTD" teaches the metadata indicating a content type (Sec. 3-5 on pp. 5-12), as it is well known that metadata is data about data and SyncML have meta-information such as parameter or attributes that are about type or content of data; therefore, metadata may be utilized for determining the content type of data.

Szeto teach a system and method comprising:

an unidentified executable (e.g. unidentified supporting application) (Fig. 12A; col. 1, ll. 55-58 and col. 12, l. 66 to col. 13, l. 16), as the received message do not identify the supporting application to be utilized with the message, and after the application type is determined from the received message, the corresponding supporting application is identified;

determine a content type (e.g. application type) from metadata (Fig. 12A; col. 1, ll. 55-58 and col. 12, l. 66 to col. 13, l. 16), in combination with SyncML Meta-Information DTD's teaching, the received message having metadata with content types including movie trailer, game, animation cartoon, advertisement, and flash presentation, and to determine the supporting application for the received message, the metadata is examining to determine the application type (e.g. content type); and automatically identifying an executable (e.g. supporting application) using the content type determined from metadata (Fig. 12A; col. 1, ll. 55-58 and col. 12, l. 66 to col. 13, l. 16), in combination with SyncML Meta-Information DTD's teaching, the received message would be associated with content types including movie trailer, game, animation cartoon, advertisement, and flash presentation in the metadata, and by using the content type, the corresponding supporting application would be identified.

It would have been obvious for one of ordinary skill in this art, at the time of invention was made to include SyncML Meta-Information DTD's content type and metadata and Szeto's identification of the executable into Rao's operation on the first identified data for the benefit of properly operating in accordance SyncML standard as in Rao's system and also for the benefit to the having a reliable system and method for a user to execute and control application (Szeto, col. 2, ll. 30-33) to obtain the invention as specified in claim 1.

14. As per claim 3, Rao, SyncML Meta-Information DTD and Szeto teach all the limitations of claim 1 as discussed above, where Rao further teaches the method comprising wherein the command contains an identifier (e.g. URI) of the first data (Rao, col. 6, l. 49 to col. 7, l. 19 and col. 8, ll. 25-34).

15. As per claim 4, Rao, SyncML Meta-Information DTD and Szeto teach all the limitations of claim 3 as discussed above, where Rao further teaches the method

comprising wherein the identifier identifies a node of a hierarchical nodular data structure (e.g. tree data structure) (Rao, col. 6, l. 49 to col. 7, l. 19 and col. 8, ll. 25-34).

16. As per claim 5, Rao, SyncML Meta-Information DTD and Szeto teach all the limitations of claim 4 as discussed above, where Rao further teaches the method comprising wherein the command is an exec command and the identifier is a uniform resource identifier contained within a source element, which is contained within the exec command (Rao, col. 6, l. 49 to col. 7, l. 19 and col. 8, l. 25 to col. 12, l. 19).

17. As per claim 6, Rao, SyncML Meta-Information DTD and Szeto teach all the limitations of claim 1 as discussed above, where Rao and Szeto further teach the method comprising wherein the command is received as extensible markup language code (Rao, col. 6, ll. 49 to col. 7, l. 3 and Szeto, col. 7, ll. 48-53).

18. As per claim 7, Rao, SyncML Meta-Information DTD and Szeto teach all the limitations of claim 6 as discussed above, where Rao further teaches the method comprising wherein the command is a SyncML command (Rao, col. 6, ll. 49 to col. 7, l. 3 and col. 8, l. 25 to col. 12, l. 19).

19. As per claim 8, Rao, SyncML Meta-Information DTD and Szeto teach all the limitations of claim 1 as discussed above, where Rao further teaches the method

comprising wherein the identified first data is stored at the electronic device (Rao, col. 3, ll. 52-63; col. 5, ll. 23-32; col. 7, ll. 38-41 and col. 11, l. 48 to col. 12, l. 19).

20. As per claim 9, Rao, SyncML Meta-Information DTD and Szeto teach all the limitations of claim 6 as discussed above, where Rao further teaches the method comprising wherein the identified first data is stored at a first leaf node of a hierarchical nodular data structure (e.g. tree data structure) (Rao, col. 3, ll. 52-63; col. 6, l. 49 to col. 7, l. 19; col. 8, ll. 25-34 and col. 11, l. 48 to col. 12, l. 19), as the data would be store in the first leaf node of the tree data structure.

21. As per claim 10, Rao, SyncML Meta-Information DTD and Szeto teach all the limitations of claim 9 as discussed above, where all further teach the method comprising wherein the metadata is associated with the first leaf node and identifies the content type of the first data stored at the first leaf node of the hierarchical data structure (e.g. tree data structure) (Rao, col. 6, l. 49 to col. 7, l. 19; col. 8, l. 25 to col. 12, l. 19, SyncML Meta-Information DTD, pp. 5-6, and Szeto, Fig. 12A; col. 1, ll. 55-58; col. 12, l. 66 to col. 13, l. 16).

22. As per claim 12, Rao, SyncML Meta-Information DTD and Szeto teach all the limitations of claim 1 as discussed above, where SyncML Meta-Information DTD and Szeto further teach the method comprising wherein determining the content type uses at least one of the value of a format element and the value of a Type element

associated with the first data (SyncML Meta-Information DTD, pp. 5-12 and Szeto, Fig. 12A; col. 1, ll. 55-58; col. 12, l. 66 to col. 13, l. 16).

23. As per claim 13, Rao, SyncML Meta-Information DTD and Szeto teach all the limitations of claim 1 as discussed above, where Szeto further teaches the method comprising associating a plurality of different executables (e.g. different supporting applications for movie trailer, game, animation cartoon, advertisement, flash presentation) with each of a plurality of different content types (Szeto, Fig. 12A; col. 1, ll. 55-58 and col. 12, l. 66 to col. 13, l. 16), as each different content types have the corresponding supporting application.

24. As per claim 14, Rao, SyncML Meta-Information DTD and Szeto teach all the limitations of claim 13 as discussed above, where SyncML Meta-Information DTD and Szeto further teach the method comprising wherein automatically identifying an executable from the content type comprises identifying the executable associated with the content type (SyncML Meta-Information DTD, pp. 5-12 and Szeto, Fig. 12A; col. 1, ll. 55-58; col. 12, l. 66 to col. 13, l. 16).

25. As per claim 15, Rao, SyncML Meta-Information DTD and Szeto teach all the limitations of claim 13 as discussed above, where Rao and Szeto further teach the method comprising wherein the plurality of different executables are stored in the electronic device (Rao, Fig 1; col. 5, l. 23 to col. 6, l. 4 and Szeto, Fig. 12A; col. 1, ll. 55-

58; col. 12, l. 66 to col. 13, l. 16), as the electronic devise would have the corresponding supporting application for operating the first data.

26. As per claim 16, Rao, SyncML Meta-Information DTD and Szeto teach all the limitations of claim 1 as discussed above, where Rao further teaches the method comprising before receiving the command specifying execution of the unidentified executable on the first data, receiving commands for creating a hierarchical nodular data structure (e.g. tree data structure) including the first data at the electronic device (Rao, col. 6, l. 49 to col. 7, l. 19 and col. 7, ll. 38-41 SyncML Meta-Information DTD, pp. 5-12), as the tree data structure is created prior to the execution of the update command.

27. As per claim 41, Rao teaches a computer program product comprising program instructions embodied on a tangible computer readable-readable medium, execution of the program instructions resulting in operations comprising:

automatically determining (e.g. determining via recognition) a property of a first data (e.g. property identifying first data to be firmware update data) (Fig. 1; col. 3, ll. 21-44; col. 6, l. 49 to col. 7, l. 19 and col. 8, l. 25 to col. 12, l. 19), as the received command is recognized by the electronic device to have the property associated with firmware updating;

utilization of metadata protocol, wherein the first data would have the corresponding metadata (col. 6, l. 49 to col. 7, l. 19); and enabling the first data to be operated on using an executable (e.g. module)(col. 5, ll. 23-32 and col. 5, l. 61 to col. 6, l. 4), as the module would operate on the firmware update data via downloading and updating processes.

Rao does not teach the computer program product comprising: determine a content type from the metadata; and automatically identifying an executable using the content type determined from the metadata for operation.

SyncML Meta-Information DTD" teaches the metadata indicating a content type (Sec. 3-5 on pp. 5-12), as it is well known that metadata is data about data and SyncML have meta-information such as parameter or attributes that are about type or content of data; therefore, metadata may be utilized for determining the content type of data.

Szeto teach a system and method comprising:

determine a content type (e.g. application type) from metadata (Fig. 12A; col. 1, ll. 55-58 and col. 12, l. 66 to col. 13, l. 16), in combination with SyncML Meta-Information DTD's teaching, the received message having metadata with content types including movie trailer, game, animation cartoon, advertisement, and flash presentation, and to determine the supporting application for the received message, the metadata is examining to determine the application type (e.g. content type); and

automatic identifying an executable (e.g. supporting application) using the content type determined from metadata for operation (Fig. 12A; col. 1, ll. 55-58 and col. 12, l. 66 to col. 13, l. 16), in combination with SyncML Meta-Information DTD's teaching, the received message would be associated with content types including movie trailer, game, animation cartoon, advertisement, and flash presentation in the metadata, and by using the content type, the corresponding supporting application would be identified.

It would have been obvious for one of ordinary skill in this art, at the time of invention was made to include Szeto's identification of the executable into Rao's operation of the first data for the benefit of having a reliable system and method for a user to execute and control application (Szeto, col. 2, ll. 30-33) to obtain the invention as specified in claim 41.

IV. CLOSING COMMENTS

Conclusion

a. STATUS OF CLAIMS IN THE APPLICATION

The following is a summary of the treatment and status of all claims in the application as recommended by **M.P.E.P. 707.07(i)**:

a(1) CLAIMS REJECTED IN THE APPLICATION

THIS ACTION IS MADE FINAL. Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the mailing date of this final action.

b. DIRECTION OF FUTURE CORRESPONDENCES

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Chun-Kuan (Mike) Lee whose telephone number is (571) 272-0671. The examiner can normally be reached on 8AM to 5PM.

IMPORTANT NOTE

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Alford Kindred can be reached on (571) 272-4037. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

/C.K.L./

June 18, 2010

Chun-Kuan (Mike) Lee
Examiner
Art Unit 2181

/Alford W. Kindred/

Supervisory Patent Examiner, Art Unit 2181